

E 5623

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-353292

(43)Date of publication of application : 24.12.1999

(51)Int.Cl.

G06F 15/16

G06F 11/20

(21)Application number : 10-160479

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 09.06.1998

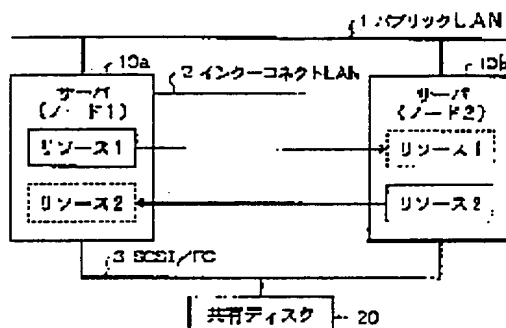
(72)Inventor : KOBAYASHI HIRONOBU

(54) CLUSTER SYSTEM AND ITS FAIL OVER CONTROL METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a cluster system which executes a proper fail over operation in accordance with the operating state of the fail over destination.

SOLUTION: In this cluster system, two server computers 10a and 10b which are connected to a public LAN 1 are loosely coupled to each other with an interconnect LAN 2. Then both computers 10a and 10b execute the communication to confirm their normal operations with each other using the LAN 2 at every prescribed interval. When one of both computers detects a failure of the other, the computer fails over the system resources operating at the other computer on its own computer. In this case, however, the control is carried out not to mechanically fail over the all system resources of the faulty computer but to change the priorities of system resources according to its own operating state and also to chance the priorities of the system resources which are originally operating at its own computer.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-353292

(43) 公開日 平成11年(1999)12月24日

(51) Int.Cl. ⁹	識別記号	F I
G 0 6 F 15/16	4 7 0	G 0 6 F 15/16
11/20	3 1 0	11/20
		4 7 0 B
		3 1 0 F

審査請求 未請求 請求項の数 4 O L (全 6 頁)

(21) 出願番号 特願平10-160479

(22) 出願日 平成10年(1998)6月9日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 小林 弘伸

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

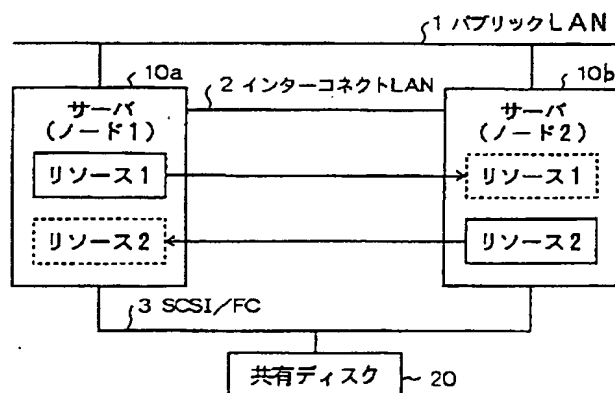
(74) 代理人 弁理士 鈴江 武彦 (外6名)

(54) 【発明の名称】 クラスタシステムおよび同システムのフェールオーバー制御方法

(57) 【要約】

【課題】フェールオーバー先の稼動状況に応じて適切なフェールオーバーを実行するクラスタシステムを提供する。

【解決手段】この発明のクラスタシステムは、パブリックLAN 1に接続された2台のサーバコンピュータ 10a~bがインターコネクトLAN 2で疎結合された構成となっており、これら双方は、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信をインターコネクトLAN 2を用いて実行し、相手の故障を検知したときに、相手のコンピュータ上で動作していたシステム資源を自身のコンピュータ上にフェールオーバーさせる。このとき、すべてを機械的にフェールオーバーさせるのではなく、自身の稼動状況に応じてその優先度を変更させ、また、自身の下で元々動作していたシステム資源の優先度を変更させるといった制御を実行する。



【特許請求の範囲】

【請求項1】 複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムにおいて、前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバー制御手段を具備することを特徴とするクラスタシステム。

【請求項2】 前記フェールオーバー制御手段は、前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御する手段を有することを特徴とする請求項1記載のクラスタシステム。

【請求項3】 複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムのフェールオーバー制御方法において、前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバー制御方法。

【請求項4】 前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を制御する請求項3記載のフェールオーバー制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、疎結合された複数のコンピュータから構成されるクラスタシステムおよび同システムのフェールオーバー制御方法に関する。

【0002】

【従来の技術】 近年、コンピュータの普及により情報化が急速に進んでおり、様々な業種でコンピュータシステムが構築されている。また、これらコンピュータシステムに寄せられる耐障害性向上の要求は、年々強まる一方である。そして、この耐障害性を実現するシステムとして、クラスタシステムが存在する。

【0003】 このクラスタシステムは、たとえば磁気ディスク装置などを共有する疎結合された複数のコンピュータから構成されるシステムであり、データ処理の負荷を分散する分散システムとして機能することに加えて、複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源（ユーティリティを含むアプリケーションプログラムなど）を他のコンピュータ上に引き継いで動作させる（これをフェールオーバーという）耐障害性システム

としても機能する。

【0004】 そして、このクラスタシステムは、比較的低位性能なコンピュータをLAN（Local Area Network）などで結合するといった機器構成で高性能かつ高信頼性を得られるために、最近ではコスト面からも注目されてきているシステムである。

【0005】 このクラスタシステムでは、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信（これをハートビートという）を実行し、相手の正常稼動を確認できなかったとき、すなわち、相手の故障を検知したときに、相手のコンピュータ上で動作していたシステム資源を自身のコンピュータ上に引き継いで実行することによって対障害性を実現する。

【0006】

【発明が解決しようとする課題】 このように、複数のコンピュータを疎結合させて互いをバックアップさせるクラスタシステムは、高性能かつ高信頼性のシステムを比較的安価に構築できるものである。

【0007】 ところで、従来のクラスタシステムにおいては、互いにハートビートを実行し合う相手と、その相手のどのシステム資源をフェールオーバーさせるかを設定するのみであった。したがって、相手のコンピュータが故障した場合には、自身のコンピュータの稼動状況に関わらずに、設定されたすべてのシステム資源がフェールオーバーされる結果、自身のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えてしまうことがあった。また、自身のコンピュータ上で元々動作していた優先度の低いシステム資源は無条件で動作し続ける結果、場合によっては、フェールオーバーされるべき優先度の高いシステム資源が起動できないことがあった。

【0008】 この発明はこのような実情に鑑みてなされたものであり、フェールオーバー先の稼動状況に応じて適切なフェールオーバーを実行するクラスタシステムおよび同システムのフェールオーバー制御方法を提供することを目的とする。

【0009】

【課題を解決するための手段】 この発明は、前述した目的を達成するために、複数のコンピュータがネットワークを介して結合され、前記複数のコンピュータの中のいずれかのコンピュータが故障したときに、そのコンピュータ上で動作していたシステム資源を他のコンピュータ上に引き継いで動作させるクラスタシステムにおいて、前記引き継がれるシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するフェールオーバー制御手段を具備したものである。

【0010】 この発明のクラスタシステムにおいては、ハートビートを実行し合う複数のコンピュータの中のいずれか一方が故障した際、その故障したコンピュータ上で動作していたシステム資源であって、フェールオーバー

させるものとして設定されたシステム資源を、従来のように、そのままの優先度で機械的にすべてフェールオーバーさせるのではなく、フェールオーバー先のコンピュータの稼動状況に応じてその優先度を変更させるため（場合によっては起動しない）、フェールオーバー先のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えることもない。

【0011】また、この発明のクラスタシステムは、前記フェールオーバー制御手段が、前記システム資源の引き継ぎを実行するときに、前記他のコンピュータ上で元々動作していたシステム資源の停止を含む優先度の変更を前記他のコンピュータの稼動状況に応じて制御するようにしたものである。

【0012】この発明のクラスタシステムにおいては、フェールオーバー先のコンピュータに故障したコンピュータからシステム資源がフェールオーバーされてきたときに、フェールオーバー先のコンピュータの稼動状況に応じてそのフェールオーバー先のコンピュータ上で元々動作していたシステム資源の優先度を変更させるため（場合によっては停止させる）、フェールオーバー先のコンピュータ上で元々動作していた優先度の低いシステム資源が無条件で動作し続けることによってフェールオーバーされるべき優先度の高いシステム資源が起動できないといった事態を引き起こすこともない。

【0013】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態を説明する。図1は、この実施形態に係るクラスタシステムの構成を示す図である。図1に示すように、この実施形態のクラスタシステムは、パブリックLAN1に接続された2台のサーバコンピュータ10a～bがインターコネクトLAN2で疎結合された構成となっており、また、この2台のサーバコンピュータ10a～bは、SCSI/FC（スカジー/ファイバーチャネル）3により接続される共有ディスク20をとともに使用してデータを共有する。

【0014】この実施形態のクラスタシステムは、パブリックLAN1を介してクライアントコンピュータから要求されるデータ処理をサーバコンピュータ10a～bで分散して実行する分散システムであり、システム資源として、正常稼動時には、サーバコンピュータ10a（ここではこちらをノード1とする）側でリソース1群が動作し、一方、サーバコンピュータ10b（ここではこちらをノード2とする）側でリソース2群が動作する。

【0015】また、このサーバコンピュータ10a～b双方は、予め定められた間隔ごとに互いの正常稼動を確認し合うための通信（ハートビート）をインターコネクトLAN2を用いて実行している。

【0016】なお、このハートビートの実行や後述するフェールオーバーの実行などは、共有ディスク20からサ

ーバコンピュータ10a～bが実装するシステムメモリにロードされ、サーバコンピュータ10a～bが実装するCPUによって実行制御されるオペレーティングシステム、あるいはこのオペレーティングシステム下で動作するプログラムによって行なわれるものである。

【0017】また、サーバコンピュータ10a～bで実行されるリソース1群およびリソース2群も、共有ディスク20上のファイルからサーバコンピュータ10a～bが実装するシステムメモリにロードされ、サーバコンピュータ10a～bが実装するCPUによって実行制御される（オペレーティングシステム下で動作する）プログラムとして構成されるものである。そして、これらリソース1群またはリソース2群のいずれかに属するすべてのシステム資源それぞれは、共有ディスク20上に格納されたリソーステーブルによってそのフェールオーバーが制御される。

【0018】図2は、この実施形態のリソーステーブルの一例を示す図である。図2に示すように、この実施形態のリソーステーブルは、「リソース名」および「フェールオーバーの優先度」の2つの項目を備えており、「リソース名」欄で示されるシステム資源それぞれのフェールオーバー実行時の取り扱いが「フェールオーバーの優先度」欄に示される。

【0019】たとえば、“ネットワーク名”および“IPアドレス”は、フェールオーバー実行時の取り扱いが“必ず実行”と設定されているが、この場合、“ネットワーク名”または“IPアドレス”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況に関わらず、常にそのままの優先度でフェールオーバーが実行される。また、“アプリ1”および“アプリ2”は、フェールオーバー実行時の取り扱いが“実行プライオリティを下げる”と設定されているが、この場合、“アプリ1”または“アプリ2”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況が予め定められた条件をオーバーしない場合にはそのままの優先度で、オーバーする場合には優先度が下げられた上でフェールオーバーが実行される。さらに、“アプリ3”は、フェールオーバー実行時の取り扱いが“フェールオーバーしない”と設定されているが、この場合、“アプリ3”が動作するサーバコンピュータが故障したときには、他方のサーバコンピュータの稼動状況に関わらず、常にフェールオーバーは実行されない。

【0020】このフェールオーバー実行時の取り扱いが“実行プライオリティを下げる”と設定されている場合に用いられる予め定められる条件は、共有ディスク20上に格納された条件テーブルによって管理されるものである。

【0021】図3は、この実施形態の条件テーブルの一例を示す図である。図3に示すように、この実施形態の

条件テーブルは、「チェック項目」および「設定値」の2つの項目を備えており、「チェック項目」欄で示される事項それぞれについて、フェールオーバー先のサーバコンピュータの満足すべき値が「設定値」欄に示される。

【0022】したがって、“アプリ1”および“アプリ2”は、フェールオーバー先のサーバコンピュータの稼動状況が、スワップファイルサイズ、CPU使用率およびメモリ使用量の各事項を満足するときはそのままの優先度でフェールオーバーが実行され、満足しないときには優先度が下げられた上でフェールオーバーが実行されることになる。

【0023】ここで、図4および図5を参照してこの実施形態のクラスタシステムの動作手順を説明する。図4は、故障したコンピュータ上で動作していたシステム資源をフェールオーバーさせる際の動作手順を説明するためのフローチャートである。

【0024】複数のコンピュータの中のいずれかのコンピュータが故障すると、この実施形態のクラスタシステムでは、その故障したコンピュータ上で動作していたシステム資源分だけ以下の処理を実行する。

【0025】まず、この実施形態のクラスタシステムでは、リソーステーブルを参照してそのシステム資源の優先度を取得し（ステップA1），“必ず実行”として設定されているかどうかをまず判定する（ステップA2）。そして，“必ず実行”として設定されていた場合には（ステップA2のYES）、そのシステム資源のフェールオーバーを実行する（ステップA3）。

【0026】一方、“必ず実行”として設定されていなかった場合には（ステップA2のNO）、続いて、“実行プライオリティを下げる”として設定されているかどうかを判定する（ステップA4）。そして、“実行プライオリティを下げる”として設定されていた場合には（ステップA4のYES）、フェールオーバー先のサーバコンピュータの稼動状況と条件テーブルとを比較し（ステップA5）、条件をオーバーしていなかった場合には（ステップA6のNO）、そのままの優先度でフェールオーバーを実行し、条件をオーバーしていた場合には（ステップA6のYES）、優先度を下げた上でフェールオーバーを実行する（ステップA7）。

【0027】なお、“実行プライオリティを下げる”として設定されていなかった場合には（ステップA4のNO），“フェールオーバーしない”として設定されているものと判定し、フェールオーバーの実行は行なわない。

【0028】また、図5は、フェールオーバー先のコンピュータ上で元々動作しているシステム資源を管理する際の動作手順を説明するためのフローチャートである。フェールオーバーが発生すると、この実施形態のクラスタシステムでは、フェールオーバー先のコンピュータ上で元々動作しているシステム資源分だけ以下の処理を実行する。

【0029】まず、この実施形態のクラスタシステムでは、リソーステーブルを参照してそのシステム資源の優先度を取得し（ステップB1），“必ず実行”として設定されているかどうかをまず判定する（ステップB2）。そして，“必ず実行”として設定されていた場合には（ステップA2のYES）、そのシステム資源については何の処理も施さずにそのまま実行を継続させる。

【0030】一方、“必ず実行”として設定されていなかった場合には（ステップB2のNO）、フェールオーバー先のサーバコンピュータの稼動状況と条件テーブルとを比較し（ステップB3）、条件をオーバーしていなかった場合には（ステップB4のNO）、そのシステム資源については何の処理も施さずにそのまま実行を継続させる。また、条件をオーバーしていた場合には（ステップB4のYES）、さらに、実行プライオリティを下げる”として設定されているかどうかを判定し（ステップB5），“実行プライオリティを下げる”として設定されていた場合には（ステップB5のYES）、優先度を下げた上で実行を継続させる（ステップB6）。一方、“実行プライオリティを下げる”として設定されていなかった場合には（ステップB5のNO），“フェールオーバーしない”として設定されているものと判定し、そのシステム資源の実行を終了する（ステップB7）。

【0031】このように、この実施形態のクラスタシステムによれば、フェールオーバー先のコンピュータの稼動状況に応じて、故障したコンピュータ上で動作していたシステム資源であってフェールオーバーを実行するものとして設定されたシステム資源と、フェールオーバー先のコンピュータ上で元々動作しているシステム資源の双方が適切に制御されることになる。

【0032】

【発明の効果】以上詳述したように、この発明によれば、ハートビートを実行し合う複数のコンピュータの中のいずれか一方が故障した際、その故障したコンピュータ上で動作していたシステム資源であって、フェールオーバーさせるものとして設定されたシステム資源を、従来のように、そのままの優先度で機械的にすべてフェールオーバーさせるのではなく、フェールオーバー先のコンピュータの稼動状況に応じてその優先度を変更させるため（場合によっては起動しない）、フェールオーバー先のコンピュータ上で元々動作していた優先度の高いシステム資源に悪影響を与えることもない。

【0033】また、フェールオーバー先のコンピュータに故障したコンピュータからシステム資源がフェールオーバーされてきたときに、フェールオーバー先のコンピュータの稼動状況に応じてそのフェールオーバー先のコンピュータ上で元々動作していたシステム資源の優先度を変更させるため（場合によっては停止させる）、フェールオーバー先のコンピュータ上で元々動作していた優先度の低いシステム資源が無条件で動作し続けることによってフェ

ールオーバーされるべき優先度の高いシステム資源が起動できないといった事態を引き起こすこともない。

【図面の簡単な説明】

【図 1】 この発明の実施形態に係るクラスタシステムの構成を示す図。

【図 2】 同実施形態のリソーステーブルの一例を示す図。

【図 3】 同実施形態の実施形態の条件テーブルの一例を示す図。

【図 4】 同実施形態の故障したコンピュータ上で動作していたシステム資源をフェールオーバーさせる際の動作手

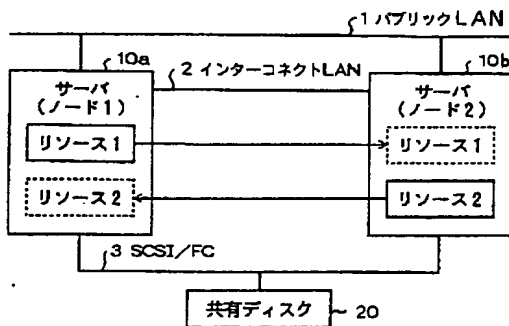
順を説明するためのフローチャート。

【図 5】 同実施形態のフェールオーバー先のコンピュータ上で元々動作しているシステム資源を管理する際の動作手順を説明するためのフローチャート。

【符号の説明】

- 1…パブリック LAN
- 2…インターコネクト LAN
- 3…SCSI/FC
- 10a～b…サーバコンピュータ
- 20…共有ディスク

【図 1】



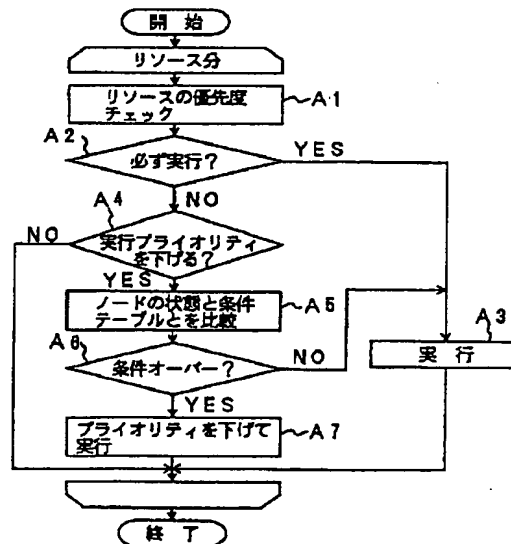
【図 2】

リソース名	フェールオーバーの優先度
ネットワーク名	必ず実行
IPアドレス	必ず実行
アプリ1	実行プライオリティを下げる
アプリ2	実行プライオリティを下げる
アプリ3	フェールオーバーしない

【図 3】

チェック項目	設定値
スワップファイルサイズ	××MB以下
CPU使用率	××%以下
メモリ使用量	××MB以下

【図 4】



【図5】

